

Capstone

September 14, 2018

```
In [1]: import numpy as np # library to handle data in a vectorized manner

import pandas as pd # library for data analysis
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # library to handle JSON files

get_ipython().system("conda install -c conda-forge geopy --yes")
from geopy.geocoders import Nominatim # convert an address into latitude and longitude via geocoding

import requests # library to handle requests
from pandas.io.json import json_normalize # transform JSON file into a pandas dataframe

# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors

# import k-means from clustering stage
from sklearn.cluster import KMeans

get_ipython().system("conda install -c conda-forge folium=0.5.0 --yes")
import folium # map rendering library

get_ipython().system('conda install -c conda-forge beautifulsoup4 --yes')
from bs4 import BeautifulSoup

get_ipython().system('conda install -c conda-forge lxml --yes')
get_ipython().system('conda install -c conda-forge html5lib --yes')
get_ipython().system('conda install -c conda-forge requests --yes')

import requests
get_ipython().system('conda install -c conda-forge geocoder --yes')
import geocoder

print('Libraries imported.')
```

```
Solving environment: done

## Package Plan ##

environment location: /home/jupyterlab/conda

added / updated specs:
- geopy
```

The following packages will be downloaded:

package	build		
geopy-1.17.0	py_0	49 KB	conda-forge
geographiclib-1.49	py_0	32 KB	conda-forge
Total:		82 KB	

The following NEW packages will be INSTALLED:

```
geographiclib: 1.49-py_0    conda-forge
geopy:          1.17.0-py_0  conda-forge
```

Downloading and Extracting Packages

```
geopy-1.17.0      | 49 KB      | #####| 100%
geographiclib-1.49 | 32 KB      | #####| 100%
```

Preparing transaction: done

Verifying transaction: done

Executing transaction: done

Solving environment: done

```
## Package Plan ##
```

```
environment location: /home/jupyterlab/conda
```

```
added / updated specs:
- folium=0.5.0
```

The following packages will be downloaded:

package	build		
folium-0.5.0	py_0	45 KB	conda-forge
altair-2.2.2	py36_1	461 KB	conda-forge
branca-0.3.0	py_0	24 KB	conda-forge

vincent-0.4.4		py_1	28 KB	conda-forge
			Total:	558 KB

The following NEW packages will be INSTALLED:

```
altair: 2.2.2-py36_1 conda-forge
branca: 0.3.0-py_0 conda-forge
folium: 0.5.0-py_0 conda-forge
vincent: 0.4.4-py_1 conda-forge
```

Downloading and Extracting Packages

folium-0.5.0	45 KB	#####	100%
altair-2.2.2	461 KB	#####	100%
branca-0.3.0	24 KB	#####	100%
vincent-0.4.4	28 KB	#####	100%

Preparing transaction: done

Verifying transaction: done

Executing transaction: done

Solving environment: done

Package Plan

```
environment location: /home/jupyterlab/conda
```

```
added / updated specs:
 - beautifulsoup4
```

The following packages will be downloaded:

package		build
beautifulsoup4-4.6.3		py36_0
		138 KB conda-forge

The following packages will be UPDATED:

```
beautifulsoup4: 4.6.0-py36h49b8c8c_1 --> 4.6.3-py36_0 conda-forge
```

Downloading and Extracting Packages

beautifulsoup4-4.6.3 138 KB	#####	100%
-------------------------------	-------	------

Preparing transaction: done

Verifying transaction: done

Executing transaction: done

Solving environment: done

```
## Package Plan ##

environment location: /home/jupyterlab/conda

added / updated specs:
- lxml
```

The following packages will be downloaded:

package	build	
lxml-4.2.5	py36hc9114bc_0	6.0 MB conda-forge

The following packages will be UPDATED:

```
lxml: 4.2.1-py36h23eabaa_0 --> 4.2.5-py36hc9114bc_0 conda-forge
```

```
Downloading and Extracting Packages
lxml-4.2.5           | 6.0 MB    | #####| 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Solving environment: done
```

```
# All requested packages already installed.
```

Solving environment: done

```
## Package Plan ##
```

```
environment location: /home/jupyterlab/conda
```

```
added / updated specs:
- requests
```

The following packages will be downloaded:

package	build	
requests-2.19.1	py36_1	94 KB conda-forge

The following packages will be UPDATED:

```
requests: 2.18.4-py36he2e5f8d_1 --> 2.19.1-py36_1 conda-forge
```

```

Downloading and Extracting Packages
requests-2.19.1      | 94 KB      | #####| 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Solving environment: done

## Package Plan ##

environment location: /home/jupyterlab/conda

added / updated specs:
- geocoder

```

The following packages will be downloaded:

package		build		
orderedset-2.0		py36_0	231 KB	conda-forge
geocoder-1.38.1		py_0	52 KB	conda-forge
ratelim-0.1.6		py36_0	5 KB	conda-forge
Total:			288 KB	

The following NEW packages will be INSTALLED:

```

geocoder: 1.38.1-py_0 conda-forge
orderedset: 2.0-py36_0 conda-forge
ratelim: 0.1.6-py36_0 conda-forge

```

```

Downloading and Extracting Packages
orderedset-2.0      | 231 KB      | #####| 100%
geocoder-1.38.1    | 52 KB       | #####| 100%
ratelim-0.1.6      | 5 KB        | #####| 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Libraries imported.

```

```

In [2]: from bs4 import BeautifulSoup
import pandas as pd
import re

headers = { 'User-Agent' : 'Mozilla/5.0' }

```

```

pageCapital = requests.get("https://en.wikipedia.org/wiki/List_of_national_capitals_by_population")
soup = BeautifulSoup(pageCapital.content, 'html.parser') # Parse the HTML as a string

# Create an object of the first object
table = soup.find("table", {"class": "wikitable sortable"})

rank=[]
country=[]
capital=[]
population=[]

for row in table.find_all('tr')[1:]:
    col=row.find_all('td')

    col1=col[0].string.strip()
    rank.append(col1)

    x = col[1].findAll('a')
    for link in x:
        col2= link.get('title')
        country.append(col2)

    x = col[2].findAll('a')
    for link in x:
        col3= link.get('title')
        capital.append(col3)

    a = str(col[3])
    start = '</span>'
    end = '<'
    col4 = (a.split(start))[1].split(end)[0]
    population.append(col4)

columns={'Rank':rank,'Country':country,'Capital':capital, 'Population': population}

# Create a dataframe from the columns variable
capitals = pd.DataFrame(columns)
capitals

```

Out[2]:

	Rank	Country \
0	1	China
1	2	India
2	3	Japan

3	4	Philippines
4	5	Russia
5	6	Egypt
6	7	Indonesia
7	8	Democratic Republic of the Congo
8	9	South Korea
9	10	Bangladesh
10	11	Mexico
11	12	Iran
12	13	England
13	14	Peru
14	15	Thailand
15	16	Colombia
16	17	Vietnam
17	18	China
18	19	Iraq
19	20	Singapore
20	21	Turkey
21	22	Chile
22	23	Saudi Arabia
23	24	Germany
24	25	Syria
25	26	Algeria
26	27	Spain
27	28	North Korea
28	29	Afghanistan
29	30	Kenya
30	31	Greece
31	32	Ethiopia
32	33	Argentina
33	34	Italy
34	35	Ukraine
35	36	Cameroon
36	37	Taiwan
37	38	Brazil
38	39	Jordan
39	40	Angola
40	41	South Africa
41	42	France
42	43	Uzbekistan
43	44	Azerbaijan
44	45	Cuba
45	46	Cambodia
46	47	Romania
47	48	Venezuela
48	49	Republic of the Congo
49	50	Morocco
50	51	Austria

51	52	Sudan
52	53	Hungary
53	54	Poland
54	55	Belarus
55	56	Uganda
56	57	Ghana
57	58	Madagascar
58	59	Lebanon
59	60	Sweden
60	61	Ecuador
61	62	Zimbabwe
62	63	Qatar
63	64	Yemen
64	65	Guinea
65	66	Malaysia
66	67	Uruguay
67	68	Zambia
68	69	Mali
69	70	Czech Republic
70	71	Haiti
71	72	Libya
72	73	Kuwait
73	74	Serbia
74	75	Dominican Republic
75	76	Somalia
76	77	Bulgaria
77	78	Belgium
78	79	Armenia
79	80	Mozambique
80	81	Sierra Leone
81	82	Republic of Ireland
82	83	Georgia (country)
83	84	Senegal
84	85	Kazakhstan
85	86	Guatemala
86	87	Liberia
87	88	Burkina Faso
88	89	Nepal
89	90	Pakistan
90	91	Canada
91	92	Nicaragua
92	93	Myanmar
93	94	Mongolia
94	95	Malawi
95	96	Bolivia
96	-	Iraqi Kurdistan
97	97	Kyrgyzstan
98	98	Togo

99	99	Panama
100	100	Netherlands
101	101	Croatia
102	102	Oman
103	103	Niger
104	104	Moldova
105	105	Israel
106	106	Nigeria
107	107	Albania
108	108	Tunisia
109	109	Turkmenistan
110	110	Chad
111	111	Honduras
112	112	Central African Republic
113	113	Mauritania
114	114	Rwanda
115	115	Latvia
116	116	Jamaica
117	117	United States
118	118	Norway
119	119	Finland
120	120	United Arab Emirates
121	121	Tajikistan
122	122	Portugal
123	123	Denmark
124	124	Lithuania
125	125	Gabon
126	126	Eritrea
127	127	El Salvador
128	128	Republic of Macedonia
129	129	Paraguay
130	130	China
131	131	Scotland
132	132	Djibouti
133	133	Ivory Coast
134	134	Guinea-Bissau
135	135	Estonia
136	136	Slovakia
137	137	United States
138	138	New Zealand
139	139	Bosnia and Herzegovina
140	140	Burundi
141	141	South Sudan
142	141	Australia
143	142	Wales
144	143	Northern Ireland
145	144	Costa Rica
146	145	Papua New Guinea

147	146	Laos
148	147	Tanzania
149	148	Lesotho
150	149	Cyprus
151	150	Slovenia
152	151	Suriname
153	152	Namibia
154	153	The Bahamas
155	154	Botswana
156	155	Benin
157	156	Kosovo
158	157	Sahrawi Arab Democratic Republic
159	158	Transnistria
160	159	Mauritius
161	160	Montenegro
162	161	Bahrain
163	162	Guyana
164	163	Cape Verde
165	164	Switzerland
166	165	Sri Lanka
167	166	Iceland
168	167	Barbados
169	168	Maldives
170	169	Bhutan
171	170	Equatorial Guinea
172	171	France
173	172	Northern Cyprus
174	173	Fiji
175	174	Swaziland
176	175	Luxembourg
177	176	Saint Lucia
178	177	United States
179	178	Comoros
180	179	Solomon Islands
181	180	East Timor
182	181	São Tomé and Príncipe
183	182	United States
184	183	Trinidad and Tobago
185	184	Republic of Artsakh
186	185	Kingdom of the Netherlands
187	186	Saint Vincent and the Grenadines
188	187	Samoa
189	188	Vanuatu
190	189	Monaco
191	190	The Gambia
192	191	Kiribati
193	192	Kingdom of the Netherlands
194	193	Seychelles

195	194	United Kingdom
196	195	United Kingdom
197	196	Brunei
198	197	United Kingdom
199	198	United Kingdom
200	199	France
201	200	State of Palestine
202	201	Marshall Islands
203	202	Andorra
204	203	Antigua and Barbuda
205	204	Tonga
206	205	Denmark
207	206	United Kingdom
208	207	Belize
209	208	Denmark
210	209	Dominica
211	210	Saint Kitts and Nevis
212	211	Finland
213	212	United States
214	213	Federated States of Micronesia
215	214	United Kingdom
216	215	Grenada
217	216	Malta
218	217	France
219	218	France
220	219	France
221	220	New Zealand
222	221	Liechtenstein
223	222	San Marino
224	223	Tuvalu
225	224	United Kingdom
226	225	United Kingdom
227	226	Norway
228	227	Australia
229	228	Kingdom of the Netherlands
230	229	France
231	230	United Kingdom
232	231	Nauru
233	232	United States
234	233	United Kingdom
235	234	United Kingdom
236	235	Australia
237	236	Vatican City
238	237	United Kingdom
239	238	New Zealand
240	239	New Zealand
241	240	Palau
242	241	Australia

243	242	United Kingdom	
244	243	United Kingdom	
		Capital	Population
0		Beijing	20,693,000
1		New Delhi	16,787,949
2		Tokyo	13,189,000
3		Metro Manila	12,877,253
4		Moscow	11,541,000
5		Cairo	10,230,350
6		Jakarta	10,187,595
7		Kinshasa	10,125,000
8		Seoul	9,989,795
9		Dhaka	8,906,000
10		Mexico City	8,851,080
11		Tehran	8,846,782
12		London	8,630,100
13		Lima	8,481,415
14		Bangkok	8,249,117
15		Bogotá	7,613,303
16		Hanoi	7,587,800
17		Hong Kong	7,298,600
18		Baghdad	7,216,040
19		Singapore	5,535,000
20		Ankara	5,150,072
21		Santiago Metropolitan Region	5,084,038
22		Riyadh	4,878,723
23		Berlin	3,520,000
24		Damascus	3,500,000
25		Algiers	3,415,811
26		Madrid	3,233,527
27		Pyongyang	3,144,005
28		Kabul	3,140,853
29		Nairobi	3,138,369
30		Athens	3,090,508
31		Addis Ababa	3,040,740
32		Buenos Aires	2,891,082
33		Rome	2,868,104
34		Kiev	2,847,200
35		Yaoundé	2,765,568
36		Taipei	2,686,516
37		Brasília	2,648,532
38		Amman	2,600,603
39		Luanda	2,453,779
40		Pretoria	2,345,908
41		Paris	2,241,346
42		Tashkent	2,207,850
43		Baku	2,204,200

44	Havana	2,135,498
45	Phnom Penh	2,011,725
46	Bucharest	1,942,254
47	Caracas	1,838,939
48	Brazzaville	1,827,000
49	Rabat	1,789,635
50	Vienna	1,749,673
51	Khartoum	1,740,661
52	Budapest	1,729,040
53	Warsaw	1,711,324
54	Minsk	1,702,061
55	Kampala	1,659,600
56	Accra	1,640,507
57	Antananarivo	1,613,375
58	Beirut	1,574,387
59	Stockholm	1,515,017
60	Quito	1,504,991
61	Harare	1,487,028
62	Doha	1,450,000
63	Sana'a	1,431,649
64	Conakry	1,399,981
65	Kuala Lumpur	1,381,830
66	Montevideo	1,369,797
67	Lusaka	1,331,254
68	Bamako	1,289,626
69	Prague	1,241,664
70	Port-au-Prince	1,235,227
71	Tripoli	1,184,045
72	Kuwait City	1,171,880
73	Belgrade	1,154,589
74	Santo Domingo	1,111,838
75	Mogadishu	1,097,133
76	Sofia	1,090,295
77	Brussels	1,080,790
78	Yerevan	1,080,487
79	Maputo	1,076,689
80	Freetown	1,070,200
81	Dublin	1,173,179
82	Tbilisi	1,044,993
83	Dakar	1,030,594
84	Astana	1,029,556
85	Guatemala City	1,022,000
86	Monrovia	1,010,970
87	Ouagadougou	1,005,231
88	Kathmandu	1,003,285
89	Islamabad	955,629
90	Ottawa	934,243
91	Managua	926,883

92	None	925,000
93	Ulaanbaatar	907,802
94	Lilongwe	902,388
95	La Paz	877,363
96	Erbil	852,500
97	Bishkek	843,240
98	Lomé	824,738
99	Panama City	813,097
100	Amsterdam	855,965
101	Zagreb	804,200
102	Muscat, Oman	797,000
103	Niamey	794,814
104	Chiinu	794,800
105	None	780,200
106	Abuja	778,567
107	Tirana	763,634
108	Tunis	767,629
109	Ashgabat	763,537
110	N'Djamena	751,288
111	Tegucigalpa	735,982
112	Bangui	731,548
113	Nouakchott	719,167
114	Kigali	718,414
115	Riga	713,016
116	Kingston, Jamaica	701,063
117	Washington, D.C.	658,893
118	Oslo	645,701
119	Helsinki	635,591
120	Abu Dhabi	585,097
121	Dushanbe	582,496
122	Lisbon	564,657
123	Copenhagen	562,253
124	Vilnius	556,723
125	Libreville	556,425
126	Asmara	1,258,001
127	San Salvador	521,366
128	Skopje	521,000
129	Asunción	520,722
130	Macau	520,400
131	Edinburgh	492,680
132	Djibouti (city)	475,332
133	Yamoussoukro	454,929
134	Bissau	452,640
135	Tallinn	440,206
136	Bratislava	424,207
137	San Juan, Puerto Rico	421,356
138	Wellington	405,000
139	Sarajevo	395,133

140	Bujumbura	384,461
141	Juba	372,410
142	Canberra	354,644
143	Cardiff	346,100
144	Belfast	333,871
145	San José, Costa Rica	328,195
146	Port Moresby	299,396
147	Vientiane	287,579
148	Dodoma	287,200
149	Maseru	267,652
150	Nicosia	270,000
151	Ljubljana	280,140
152	Paramaribo	254,147
153	Windhoek	252,721
154	Nassau, Bahamas	248,948
155	Gaborone	225,656
156	Porto-Novo	223,552
157	Prishtina	198,214
158	Tifariti	194,668
159	Tiraspol	159,163
160	Port Louis	147,251
161	Podgorica	141,854
162	Manama	140,616
163	Georgetown, Guyana	134,599
164	Praia	125,464
165	De facto	121,631
166	Colombo	118,556
167	Reykjavík	115,000
168	Bridgetown	110,000
169	Malé	103,693
170	Thimphu	101,259
171	Malabo	100,677
172	Nouméa	89,207
173	Nicosia	84,893
174	Suva	84,410
175	Mbabane	81,594
176	Luxembourg (city)	76,420
177	Castries	70,000
178	Saipan	62,392 (2000)
179	Moroni, Comoros	60,200
180	Honiara	59,288
181	Dili	59,069
182	São Tomé	56,166
183	Pago Pago	52,000 (2003)
184	Port of Spain	50,479
185	Stepanakert	49,986
186	Willemstad	49,885
187	Kingstown	40,020

188	Apia	39,813
189	Port Vila	38,000
190	Monaco	35,986
191	Banjul	34,828
192	Tarawa	30,000
193	Oranjestad, Aruba	29,998
194	Victoria, Seychelles	29,298
195	Gibraltar	29,286
196	Saint Helier	28,380
197	Bandar Seri Begawan	28,135
198	George Town, Cayman Islands	26,798
199	Douglas, Isle of Man	26,600
200	Papeete	26,200
201	De facto	25,500
202	Majuro	25,400
203	Andorra la Vella	22,884
204	St. John's, Antigua and Barbuda	22,679
205	Nukualofa	22,400
206	Tórshavn	18,573
207	St. Peter Port	16,701
208	Belmopan	16,451
209	Nuuk	15,469
210	Roseau	14,847
211	Basseterre	13,043
212	Mariehamn	11,296
213	Charlotte Amalie, United States Virgin Islands	10,817
214	Palikir	9,900
215	Road Town	9,400
216	St. George's, Grenada	7,500
217	Valletta	6,444
218	Gustavia, Saint Barthélemy	6,000
219	Marigot, Saint Martin	5,700
220	Saint-Pierre, Saint Pierre and Miquelon	5,509
221	Avarua	5,445
222	Vaduz	5,248
223	City of San Marino	4,493
224	Funafuti	4,492
225	Cockburn Town	3,700
226	Stanley, Falkland Islands	2,115
227	Longyearbyen	2,075
228	Flying Fish Cove	1,493
229	Philipsburg, Sint Maarten	1,338
230	Mata-Utu	1,191
231	The Valley, Anguilla	1,169
232	De facto	1,100
233	Hagåtña	1,100
234	De facto	391
235	Hamilton, Bermuda	1,010

236	Kingston, Norfolk Island	880
237	City-state	826 (2009 est.)
238	Jamestown, Saint Helena	714
239	Alofi	616
240	Atafu	524
241	Ngerulmud	391
242	West Island, Cocos (Keeling) Islands	120
243	Adamstown, Pitcairn Islands	56
244	King Edward Point	18

```
In [3]: from bs4 import BeautifulSoup
import pandas as pd
import re

headers = { 'User-Agent' : 'Mozilla/5.0' }
pageCapital = requests.get("https://www.cia.gov/library/publications/the-world-factbook")
soup = BeautifulSoup(pageCapital.content, 'html.parser')# Parse the HTML as a string

x = re.split("( )",str(soup))
x = list(filter(lambda a: a != ' ', x))
x = list(filter(lambda a: a != '\n', x))
x = {'names' : x}
df = pd.DataFrame(x)
df = df[df.names.str.contains("\r\n") == False]

countries = []
GDP = []

for value in range(df.shape[0]):
    if value %2 == 1:
        countries.append(df.iloc[value,0])

for value in range(df.shape[0]):
    if value %2 == 0:
        GDP.append(df.iloc[value,0])

table = {'Country' : countries, 'GDP' : GDP}
GDP.pop(0)
dfGDP = pd.DataFrame(table)
dfGDP
```

```
Out[3]:
```

	Country	GDP
0	Liechtenstein	\$139,100
1	Qatar	\$124,500
2	Monaco	\$115,700

3	Macau	\$111,600
4	Luxembourg	\$106,300
5	Bermuda	\$99,400
6	Singapore	\$93,900
7	Isle of Man	\$84,600
8	Brunei	\$78,200
9	Ireland	\$75,500
10	Norway	\$71,800
11	Falkland Islands (Islas Malvinas)	\$70,800
12	United Arab Emirates	\$67,700
13	Sint Maarten	\$66,800
14	Kuwait	\$66,200
15	Gibraltar	\$61,700
16	Hong Kong	\$61,400
17	Switzerland	\$61,400
18	United States	\$59,500
19	San Marino	\$58,600
20	Jersey	\$56,600
21	Saudi Arabia	\$54,800
22	Netherlands	\$53,600
23	Guernsey	\$52,500
24	Iceland	\$51,800
25	Sweden	\$51,500
26	Germany	\$50,400
27	Taiwan	\$50,300
28	Australia	\$50,300
29	Austria	\$49,900
30	Denmark	\$49,900
31	Andorra	\$49,900
32	Bahrain	\$48,500
33	Canada	\$48,300
34	Belgium	\$46,600
35	Saint Pierre and Miquelon	\$46,200
36	Oman	\$45,200
37	Finland	\$44,300
38	United Kingdom	\$44,100
39	Cayman Islands	\$43,800
40	France	\$43,800
41	Japan	\$42,800
42	Malta	\$42,000
43	Greenland	\$41,800
44	European Union	\$40,900
45	Faroe Islands	\$40,000
46	Korea, South	\$39,400
47	New Zealand	\$38,900
48	Spain	\$38,300
49	Italy	\$38,100
50	Puerto Rico	\$37,300

51	Virgin Islands	\$37,000
52	Cyprus	\$37,000
53	Israel	\$36,300
54	Equatorial Guinea	\$36,000
55	Guam	\$35,600
56	Czechia	\$35,500
57	Slovenia	\$34,400
58	British Virgin Islands	\$34,200
59	Montserrat	\$34,000
60	Slovakia	\$33,000
61	Lithuania	\$32,300
62	Estonia	\$31,800
63	Trinidad and Tobago	\$31,400
64	Bahamas, The	\$31,200
65	New Caledonia	\$31,100
66	Portugal	\$30,400
67	Hungary	\$29,500
68	Poland	\$29,500
69	Turks and Caicos Islands	\$29,100
70	Malaysia	\$29,000
71	Seychelles	\$28,900
72	Russia	\$27,800
73	Greece	\$27,700
74	Latvia	\$27,600
75	Turkey	\$26,900
76	Saint Kitts and Nevis	\$26,800
77	Kazakhstan	\$26,300
78	Antigua and Barbuda	\$26,300
79	Panama	\$25,400
80	Aruba	\$25,300
81	Northern Mariana Islands	\$24,500
82	Chile	\$24,500
83	Romania	\$24,500
84	Croatia	\$24,400
85	Uruguay	\$22,400
86	Bulgaria	\$21,700
87	Mauritius	\$21,600
88	Argentina	\$20,900
89	Iran	\$20,200
90	Mexico	\$19,900
91	Lebanon	\$19,400
92	Saint Martin	\$19,300
93	Gabon	\$19,200
94	Maldives	\$19,100
95	Belarus	\$18,900
96	Barbados	\$18,700
97	Turkmenistan	\$18,100
98	Thailand	\$17,900

99		Botswana	\$17,800
100		Montenegro	\$17,700
101		Azerbaijan	\$17,500
102		French Polynesia	\$17,000
103		Iraq	\$17,000
104		Costa Rica	\$16,900
105		Dominican Republic	\$16,900
106		Cook Islands	\$16,700
107		China	\$16,700
108		Palau	\$16,200
109		Brazil	\$15,600
110		Algeria	\$15,200
111		Serbia	\$15,000
112		Curacao	\$15,000
113		Macedonia	\$14,900
114		Grenada	\$14,900
115		Suriname	\$14,600
116		Colombia	\$14,500
117		Saint Lucia	\$14,400
118		South Africa	\$13,500
119		Peru	\$13,300
120		Mongolia	\$13,000
121		Sri Lanka	\$12,800
122		Egypt	\$12,700
123		Bosnia and Herzegovina	\$12,700
124		Albania	\$12,500
125		Jordan	\$12,500
126		Indonesia	\$12,400
127		Cuba	\$12,300
128		Nauru	\$12,200
129		Anguilla	\$12,200
130		Venezuela	\$12,100
131		Tunisia	\$11,800
132		Ecuador	\$11,500
133		Saint Vincent and the Grenadines	\$11,500
134		Namibia	\$11,300
135		American Samoa	\$11,200
136		Dominica	\$11,100
137		Georgia	\$10,700
138		Kosovo	\$10,500
139		Libya	\$10,000
140		Eswatini	\$9,900
141		Paraguay	\$9,800
142		Fiji	\$9,800
143		Armenia	\$9,500
144		Jamaica	\$9,200
145		El Salvador	\$8,900
146		Bhutan	\$8,700

147	Ukraine	\$8,700
148	Morocco	\$8,600
149	Belize	\$8,300
150	Philippines	\$8,300
151	Guyana	\$8,200
152	Guatemala	\$8,100
153	Saint Helena, Ascension, and Tristan da Cunha	\$7,800
154	Bolivia	\$7,500
155	Laos	\$7,400
156	India	\$7,200
157	Cabo Verde	\$6,900
158	Vietnam	\$6,900
159	Uzbekistan	\$6,900
160	Angola	\$6,800
161	Congo, Republic of the	\$6,600
162	Burma	\$6,200
163	Nigeria	\$5,900
164	Nicaragua	\$5,800
165	Niue	\$5,800
166	Samoa	\$5,700
167	Moldova	\$5,700
168	Tonga	\$5,600
169	Honduras	\$5,600
170	Timor-Leste	\$5,400
171	Pakistan	\$5,400
172	Ghana	\$4,700
173	Sudan	\$4,600
174	Mauritania	\$4,400
175	West Bank	\$4,300
176	Bangladesh	\$4,200
177	Zambia	\$4,000
178	Cambodia	\$4,000
179	Cote d'Ivoire	\$3,900
180	Wallis and Futuna	\$3,800
181	Tuvalu	\$3,800
182	Cameroon	\$3,700
183	Papua New Guinea	\$3,700
184	Kyrgyzstan	\$3,700
185	Lesotho	\$3,600
186	Djibouti	\$3,600
187	Kenya	\$3,500
188	Marshall Islands	\$3,400
189	Micronesia, Federated States of	\$3,400
190	Tajikistan	\$3,200
191	Tanzania	\$3,200
192	Sao Tome and Principe	\$3,200
193	Syria	\$2,900
194	Vanuatu	\$2,700

195		Senegal	\$2,700
196		Nepal	\$2,700
197	Western Sahara	\$2,500	
198		Uganda	\$2,400
199		Benin	\$2,300
200		Zimbabwe	\$2,300
201		Chad	\$2,300
202	Solomon Islands	\$2,200	
203		Mali	\$2,200
204		Ethiopia	\$2,200
205		Rwanda	\$2,100
206	Afghanistan	\$2,000	
207		Kiribati	\$2,000
208		Guinea	\$2,000
209	Burkina Faso	\$1,900	
210	Guinea-Bissau	\$1,800	
211		Haiti	\$1,800
212	Gambia, The	\$1,700	
213		Togo	\$1,700
214	Korea, North	\$1,700	
215		Comoros	\$1,600
216	Sierra Leone	\$1,600	
217		Madagascar	\$1,600
218	Eritrea	\$1,600	
219	South Sudan	\$1,500	
220		Liberia	\$1,400
221		Yemen	\$1,300
222		Malawi	\$1,200
223		Niger	\$1,200
224		Mozambique	\$1,200
225		Tokelau	\$1,000
226	Congo, Democratic Republic of the		\$800
227	Central African Republic		\$700
228		Burundi	\$700

```
In [4]: dfinal = capitals.merge(dfGDP, on="Country", how = 'inner')
dfinal = dfinal[dfinal.Capital != 'Hong Kong']
dfinal = dfinal[dfinal.Capital != 'Macau']
dfinal
```

Out [4]:	Rank	Country	Capital	Population	\
	0	China	Beijing	20,693,000	
	3	India	New Delhi	16,787,949	
	4	Philippines	Metro Manila	12,877,253	
	5	Egypt	Cairo	10,230,350	
	6	Indonesia	Jakarta	10,187,595	
	7	Bangladesh	Dhaka	8,906,000	
	8	Peru	Lima	8,481,415	

9	16	Colombia	Bogotá	7,613,303
10	17	Vietnam	Hanoi	7,587,800
11	19	Iraq	Baghdad	7,216,040
12	20	Singapore	Singapore	5,535,000
13	25	Syria	Damascus	3,500,000
14	26	Algeria	Algiers	3,415,811
15	29	Afghanistan	Kabul	3,140,853
16	30	Kenya	Nairobi	3,138,369
17	32	Ethiopia	Addis Ababa	3,040,740
18	35	Ukraine	Kiev	2,847,200
19	36	Cameroon	Yaoundé	2,765,568
20	38	Brazil	Brasília	2,648,532
21	39	Jordan	Amman	2,600,603
22	40	Angola	Luanda	2,453,779
23	41	South Africa	Pretoria	2,345,908
24	43	Uzbekistan	Tashkent	2,207,850
25	44	Azerbaijan	Baku	2,204,200
26	45	Cuba	Havana	2,135,498
27	46	Cambodia	Phnom Penh	2,011,725
28	48	Venezuela	Caracas	1,838,939
29	50	Morocco	Rabat	1,789,635
30	52	Sudan	Khartoum	1,740,661
31	56	Uganda	Kampala	1,659,600
32	57	Ghana	Accra	1,640,507
33	58	Madagascar	Antananarivo	1,613,375
34	61	Ecuador	Quito	1,504,991
35	62	Zimbabwe	Harare	1,487,028
36	63	Qatar	Doha	1,450,000
37	64	Yemen	Sana'a	1,431,649
38	65	Guinea	Conakry	1,399,981
39	68	Zambia	Lusaka	1,331,254
40	69	Mali	Bamako	1,289,626
41	71	Haiti	Port-au-Prince	1,235,227
42	72	Libya	Tripoli	1,184,045
43	74	Serbia	Belgrade	1,154,589
44	75	Dominican Republic	Santo Domingo	1,111,838
45	79	Armenia	Yerevan	1,080,487
46	80	Mozambique	Maputo	1,076,689
47	81	Sierra Leone	Freetown	1,070,200
48	84	Senegal	Dakar	1,030,594
49	86	Guatemala	Guatemala City	1,022,000
50	87	Liberia	Monrovia	1,010,970
51	88	Burkina Faso	Ouagadougou	1,005,231
52	89	Nepal	Kathmandu	1,003,285
53	90	Pakistan	Islamabad	955,629
54	92	Nicaragua	Managua	926,883
55	94	Mongolia	Ulaanbaatar	907,802
56	95	Malawi	Lilongwe	902,388

57	96	Bolivia	La Paz	877,363
58	97	Kyrgyzstan	Bishkek	843,240
59	98	Togo	Lomé	824,738
60	103	Niger	Niamey	794,814
61	104	Moldova	Chiinu	794,800
62	106	Nigeria	Abuja	778,567
63	107	Albania	Tirana	763,634
64	108	Tunisia	Tunis	767,629
65	110	Chad	N'Djamena	751,288
66	111	Honduras	Tegucigalpa	735,982
67	112	Central African Republic	Bangui	731,548
68	113	Mauritania	Nouakchott	719,167
69	114	Rwanda	Kigali	718,414
70	116	Jamaica	Kingston, Jamaica	701,063
71	121	Tajikistan	Dushanbe	582,496
72	126	Eritrea	Asmara	1,258,001
73	127	El Salvador	San Salvador	521,366
74	129	Paraguay	Asunción	520,722
75	132	Djibouti	Djibouti (city)	475,332
76	134	Guinea-Bissau	Bissau	452,640
77	139	Bosnia and Herzegovina	Sarajevo	395,133
78	140	Burundi	Bujumbura	384,461
79	141	South Sudan	Juba	372,410
80	144	Costa Rica	San José, Costa Rica	328,195
81	145	Papua New Guinea	Port Moresby	299,396
82	146	Laos	Vientiane	287,579
83	147	Tanzania	Dodoma	287,200
84	148	Lesotho	Maseru	267,652
85	151	Suriname	Paramaribo	254,147
86	152	Namibia	Windhoek	252,721
87	154	Botswana	Gaborone	225,656
88	155	Benin	Porto-Novo	223,552
89	156	Kosovo	Prishtina	198,214
90	160	Montenegro	Podgorica	141,854
91	162	Guyana	Georgetown, Guyana	134,599
92	165	Sri Lanka	Colombo	118,556
93	169	Bhutan	Thimphu	101,259
94	173	Fiji	Suva	84,410
95	175	Luxembourg	Luxembourg (city)	76,420
96	176	Saint Lucia	Castries	70,000
97	178	Comoros	Moroni, Comoros	60,200
98	179	Solomon Islands	Honiara	59,288
99	186	Saint Vincent and the Grenadines	Kingstown	40,020
100	187	Samoa	Apia	39,813
101	188	Vanuatu	Port Vila	38,000
102	189	Monaco	Monaco	35,986
103	191	Kiribati	Tarawa	30,000
104	196	Brunei	Bandar Seri Begawan	28,135

105	201	Marshall Islands	Majuro	25,400
106	204	Tonga	Nukualofa	22,400
107	207	Belize	Belmopan	16,451
108	209	Dominica	Roseau	14,847
109	215	Grenada	St. George's, Grenada	7,500
110	221	Liechtenstein	Vaduz	5,248
111	223	Tuvalu	Funafuti	4,492
112	231	Nauru	De facto	1,100
113	240	Palau	Ngerulmud	391

	GDP
0	\$16,700
3	\$7,200
4	\$8,300
5	\$12,700
6	\$12,400
7	\$4,200
8	\$13,300
9	\$14,500
10	\$6,900
11	\$17,000
12	\$93,900
13	\$2,900
14	\$15,200
15	\$2,000
16	\$3,500
17	\$2,200
18	\$8,700
19	\$3,700
20	\$15,600
21	\$12,500
22	\$6,800
23	\$13,500
24	\$6,900
25	\$17,500
26	\$12,300
27	\$4,000
28	\$12,100
29	\$8,600
30	\$4,600
31	\$2,400
32	\$4,700
33	\$1,600
34	\$11,500
35	\$2,300
36	\$124,500
37	\$1,300
38	\$2,000

39	\$4,000
40	\$2,200
41	\$1,800
42	\$10,000
43	\$15,000
44	\$16,900
45	\$9,500
46	\$1,200
47	\$1,600
48	\$2,700
49	\$8,100
50	\$1,400
51	\$1,900
52	\$2,700
53	\$5,400
54	\$5,800
55	\$13,000
56	\$1,200
57	\$7,500
58	\$3,700
59	\$1,700
60	\$1,200
61	\$5,700
62	\$5,900
63	\$12,500
64	\$11,800
65	\$2,300
66	\$5,600
67	\$700
68	\$4,400
69	\$2,100
70	\$9,200
71	\$3,200
72	\$1,600
73	\$8,900
74	\$9,800
75	\$3,600
76	\$1,800
77	\$12,700
78	\$700
79	\$1,500
80	\$16,900
81	\$3,700
82	\$7,400
83	\$3,200
84	\$3,600
85	\$14,600
86	\$11,300

```
87      $17,800
88      $2,300
89      $10,500
90      $17,700
91      $8,200
92      $12,800
93      $8,700
94      $9,800
95      $106,300
96      $14,400
97      $1,600
98      $2,200
99      $11,500
100     $5,700
101     $2,700
102     $115,700
103     $2,000
104     $78,200
105     $3,400
106     $5,600
107     $8,300
108     $11,100
109     $14,900
110     $139,100
111     $3,800
112     $12,200
113     $16,200
```

```
In [5]: dfinal['longitude'] = np.zeros(len(dfinal['Country']))
dfinal['latitude'] = np.zeros(len(dfinal['Country']))
```

```
In [6]: def get_google_results(address, api_key=None, return_full_response=False):
    # Set up your Geocoding url
    geocode_url = "https://maps.googleapis.com/maps/api/geocode/json?address={}".format(address)
    if api_key is not None:
        geocode_url = geocode_url + "&key={}".format(api_key)

    # Ping google for the results:
    results = requests.get(geocode_url)
    # Results will be in JSON format - convert to dict using requests functionality
    results = results.json()

    # if there's no results or an error, return empty results.
    if len(results['results']) == 0:
        output = {
            "latitude": None,
            "longitude": None
        }
```

```

    else:
        answer = results['results'][0]
        output = {
            "latitude": answer.get('geometry').get('location').get('lat'),
            "longitude": answer.get('geometry').get('location').get('lng'),
        }
    return output

In [7]: RETURN_FULL_RESULTS = False

for x in range(112):
    adress = dfinal.iloc[x,2]+','+dfinal.iloc[x,1]
    result = geocode_result = get_google_results(adress, API_KEY, return_full_response=RETURN_FULL_RESPONSE)
    dfinal.iloc[x,5] = result['longitude']
    dfinal.iloc[x,6] = result['latitude']

In [8]: dfinal = dfinal.drop('Rank', axis=1)

In [26]: newdf = dfinal.sample(n=100)
         newdf['Number of Hospitals'] = np.zeros(len(newdf['Country']))

In [28]: for x in range(newdf.shape[0]):
         longitude = newdf.iloc[x,4]
         latitude = newdf.iloc[x,5]
         LIMIT = 100
         radius = 5000
         url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&ll={},{}&radius={}&limit={}'.format(CLIENT_ID, CLIENT_SECRET, latitude, longitude, radius, LIMIT)
         results = requests.get(url).json()
         venues = results['response']['groups'][0]['items']
         nearby_venues = json_normalize(venues)
         newdf.iloc[x,6] = nearby_venues.shape[0]

```

KeyError

Traceback (most recent call last)

```

<ipython-input-28-74d0f12a7274> in <module>()
  13     LIMIT)
  14     results = requests.get(url).json()
--> 15     venues = results['response']['groups'][0]['items']
  16     nearby_venues = json_normalize(venues)

```

```
17     newdf.iloc[x,6] = nearby_venues.shape[0]
```

```
KeyError: 'groups'
```

```
In [29]: newdf['Number of Hospitals'] = newdf['Number of Hospitals'].replace('0', '1')
newdf['GDP'] = newdf['GDP'].str.replace(' ', '')
newdf['GDP'] = newdf['GDP'].str.replace('$', '')
newdf['GDP'] = newdf['GDP'].str.replace(',', '')
newdf['GDP']
newdf['GDP'] = newdf[['GDP']].apply(pd.to_numeric)
newdf['Hospitals/GDP'] = newdf['Number of Hospitals']/newdf['GDP']
```

```
In [30]: newdf['Hospitals/GDP'] = newdf['Hospitals/GDP']*1000
newdf
```

```
Out[30]:
```

	Country	Capital	Population	\
47	Sierra Leone	Freetown	1,070,200	
42	Libya	Tripoli	1,184,045	
14	Algeria	Algiers	3,415,811	
91	Guyana	Georgetown, Guyana	134,599	
111	Tuvalu	Funafuti	4,492	
107	Belize	Belmopan	16,451	
60	Niger	Niamey	794,814	
93	Bhutan	Thimphu	101,259	
45	Armenia	Yerevan	1,080,487	
74	Paraguay	Asunción	520,722	
92	Sri Lanka	Colombo	118,556	
20	Brazil	Brasília	2,648,532	
102	Monaco	Monaco	35,986	
77	Bosnia and Herzegovina	Sarajevo	395,133	
26	Cuba	Havana	2,135,498	
83	Tanzania	Dodoma	287,200	
39	Zambia	Lusaka	1,331,254	
71	Tajikistan	Dushanbe	582,496	
88	Benin	Porto-Novo	223,552	
85	Suriname	Paramaribo	254,147	
56	Malawi	Lilongwe	902,388	
72	Eritrea	Asmara	1,258,001	
66	Honduras	Tegucigalpa	735,982	
37	Yemen	Sana'a	1,431,649	
0	China	Beijing	20,693,000	
30	Sudan	Khartoum	1,740,661	
28	Venezuela	Caracas	1,838,939	
16	Kenya	Nairobi	3,138,369	
100	Samoa	Apia	39,813	
46	Mozambique	Maputo	1,076,689	

67	Central African Republic	Bangui	731,548
7	Bangladesh	Dhaka	8,906,000
97	Comoros	Moroni, Comoros	60,200
104	Brunei	Bandar Seri Begawan	28,135
40	Mali	Bamako	1,289,626
9	Colombia	Bogotá	7,613,303
79	South Sudan	Juba	372,410
11	Iraq	Baghdad	7,216,040
69	Rwanda	Kigali	718,414
63	Albania	Tirana	763,634
75	Djibouti	Djibouti (city)	475,332
54	Nicaragua	Managua	926,883
113	Palau	Ngerulmud	391
50	Liberia	Monrovia	1,010,970
59	Togo	Lomé	824,738
109	Grenada	St. George's, Grenada	7,500
96	Saint Lucia	Castries	70,000
62	Nigeria	Abuja	778,567
76	Guinea-Bissau	Bissau	452,640
99	Saint Vincent and the Grenadines	Kingstown	40,020
44	Dominican Republic	Santo Domingo	1,111,838
4	Philippines	Metro Manila	12,877,253
13	Syria	Damascus	3,500,000
35	Zimbabwe	Harare	1,487,028
17	Ethiopia	Addis Ababa	3,040,740
58	Kyrgyzstan	Bishkek	843,240
78	Burundi	Bujumbura	384,461
31	Uganda	Kampala	1,659,600
15	Afghanistan	Kabul	3,140,853
27	Cambodia	Phnom Penh	2,011,725
106	Tonga	Nukualofa	22,400
86	Namibia	Windhoek	252,721
25	Azerbaijan	Baku	2,204,200
89	Kosovo	Prishtina	198,214
3	India	New Delhi	16,787,949
48	Senegal	Dakar	1,030,594
105	Marshall Islands	Majuro	25,400
87	Botswana	Gaborone	225,656
5	Egypt	Cairo	10,230,350
36	Qatar	Doha	1,450,000
80	Costa Rica	San José, Costa Rica	328,195
101	Vanuatu	Port Vila	38,000
108	Dominica	Roseau	14,847
112	Nauru	De facto	1,100
51	Burkina Faso	Ouagadougou	1,005,231
61	Moldova	Chiinu	794,800
21	Jordan	Amman	2,600,603
110	Liechtenstein	Vaduz	5,248

33		Madagascar	Antananarivo	1,613,375
12		Singapore	Singapore	5,535,000
103		Kiribati	Tarawa	30,000
53		Pakistan	Islamabad	955,629
84		Lesotho	Maseru	267,652
23		South Africa	Pretoria	2,345,908
52		Nepal	Kathmandu	1,003,285
24		Uzbekistan	Tashkent	2,207,850
55		Mongolia	Ulaanbaatar	907,802
65		Chad	N'Djamena	751,288
64		Tunisia	Tunis	767,629
22		Angola	Luanda	2,453,779
18		Ukraine	Kiev	2,847,200
49		Guatemala	Guatemala City	1,022,000
57		Bolivia	La Paz	877,363
10		Vietnam	Hanoi	7,587,800
94		Fiji	Suva	84,410
19		Cameroon	Yaoundé	2,765,568
38		Guinea	Conakry	1,399,981
70		Jamaica	Kingston, Jamaica	701,063
95		Luxembourg	Luxembourg (city)	76,420
98		Solomon Islands	Honiara	59,288

	GDP	longitude	latitude	Number of Hospitals	Hospitals/GDP
47	1600	-13.231723	8.465677	0.0	0.000000
42	10000	13.191338	32.887209	4.0	0.400000
14	15200	3.058756	36.753768	8.0	0.526316
91	8200	-58.155125	6.801279	4.0	0.487805
111	3800	179.196193	-8.521147	1.0	0.263158
107	8300	-88.759020	17.251011	1.0	0.120482
60	1200	2.125385	13.511596	2.0	1.666667
93	8700	89.639286	27.472792	4.0	0.459770
45	9500	44.499103	40.179186	17.0	1.789474
74	9800	-57.575926	-25.263740	95.0	9.693878
92	12800	79.861243	6.927079	58.0	4.531250
20	15600	-47.921820	-15.826691	64.0	4.102564
102	115700	7.424616	43.738418	4.0	0.034572
77	12700	18.413076	43.856259	6.0	0.472441
26	12300	-82.366596	23.113592	6.0	0.487805
83	3200	35.751607	-6.162959	4.0	1.250000
39	4000	28.322816	-15.387526	5.0	1.250000
71	3200	68.787038	38.559772	11.0	3.437500
88	2300	2.628852	6.496857	2.0	0.869565
85	14600	-55.203828	5.852036	4.0	0.273973
56	1200	33.774119	-13.962612	5.0	4.166667
72	1600	38.925052	15.322877	0.0	0.000000
66	5600	-87.192136	14.072275	25.0	4.464286
37	1300	44.191007	15.369445	2.0	1.538462

0	16700	116.407396	39.904200	33.0	1.976048
30	4600	32.559899	15.500654	8.0	1.739130
28	12100	-66.903606	10.480594	42.0	3.471074
16	3500	36.821946	-1.292066	56.0	16.000000
100	5700	-171.751355	-13.850696	3.0	0.526316
46	1200	32.573175	-25.969248	9.0	7.500000
67	700	18.558190	4.394673	2.0	2.857143
7	4200	90.412518	23.810332	7.0	1.666667
97	1600	43.247315	-11.717216	0.0	0.000000
104	78200	114.939821	4.903052	40.0	0.511509
40	2200	-8.002889	12.639232	2.0	0.909091
9	14500	-74.072092	4.710989	41.0	2.827586
79	1500	31.571250	4.859363	2.0	1.333333
11	17000	44.361488	33.312806	4.0	0.235294
69	2100	30.104429	-1.970579	2.0	0.952381
63	12500	19.818698	41.327546	4.0	0.320000
75	3600	43.145648	11.572076	1.0	0.277778
54	5800	-86.236174	12.114993	21.0	3.620690
113	16200	134.624289	7.500383	0.0	0.000000
50	1400	-10.807370	6.315607	2.0	1.428571
59	1700	1.225418	6.125626	3.0	1.764706
109	14900	-61.748800	12.056097	1.0	0.067114
96	14400	-60.987469	14.010109	5.0	0.347222
62	5900	7.398574	9.076479	0.0	0.000000
76	1800	-15.584323	11.863220	0.0	0.000000
99	11500	-61.224816	13.160025	0.0	0.000000
44	16900	-69.931212	18.486058	0.0	0.000000
4	8300	121.022256	14.609054	0.0	0.000000
13	2900	36.276528	33.513807	0.0	0.000000
35	2300	31.033510	-17.825166	0.0	0.000000
17	2200	38.757761	8.980603	0.0	0.000000
58	3700	74.569762	42.874621	0.0	0.000000
78	700	29.359878	-3.361378	0.0	0.000000
31	2400	32.582520	0.347596	0.0	0.000000
15	2000	69.207486	34.555349	0.0	0.000000
27	4000	104.928210	11.556374	0.0	0.000000
106	5600	-175.204947	-21.139342	0.0	0.000000
86	11300	17.065755	-22.560881	0.0	0.000000
25	17500	49.867092	40.409262	0.0	0.000000
89	10500	21.165503	42.662914	0.0	0.000000
3	7200	77.209021	28.613939	0.0	0.000000
48	2700	-17.467686	14.716677	0.0	0.000000
105	3400	171.185774	7.116421	0.0	0.000000
87	17800	25.923147	-24.628208	0.0	0.000000
5	12700	31.235712	30.044420	0.0	0.000000
36	124500	51.531040	25.285447	0.0	0.000000
80	16900	-84.090725	9.928069	0.0	0.000000
101	2700	168.327325	-17.733251	0.0	0.000000

108	11100	-61.379355	15.309168	0.0	0.000000
112	12200	166.931503	-0.522778	0.0	0.000000
51	1900	-1.519660	12.371428	0.0	0.000000
61	5700	28.863810	47.010453	0.0	0.000000
21	12500	35.928372	31.945367	0.0	0.000000
110	139100	9.520928	47.141030	0.0	0.000000
33	1600	47.507905	-18.879190	0.0	0.000000
12	93900	103.819836	1.352083	0.0	0.000000
103	2000	172.971662	1.451817	0.0	0.000000
53	5400	73.047885	33.684420	0.0	0.000000
84	3600	27.486923	-29.315077	0.0	0.000000
23	13500	28.229271	-25.747868	0.0	0.000000
52	2700	85.323960	27.717245	0.0	0.000000
24	6900	69.240073	41.299496	0.0	0.000000
55	13000	106.905744	47.886399	0.0	0.000000
65	2300	15.055741	12.134846	0.0	0.000000
64	11800	10.181532	36.806495	0.0	0.000000
22	6800	13.230176	-8.814656	0.0	0.000000
18	8700	30.523400	50.450100	0.0	0.000000
49	8100	-90.506882	14.634915	0.0	0.000000
57	7500	-68.119294	-16.489689	0.0	0.000000
10	6900	105.834160	21.027764	0.0	0.000000
94	9800	178.450079	-18.124809	0.0	0.000000
19	3700	11.502075	3.848032	0.0	0.000000
38	2000	-13.578401	9.641185	0.0	0.000000
70	9200	-76.809904	18.017874	0.0	0.000000
95	106300	6.131935	49.611621	0.0	0.000000
98	2200	159.972900	-9.445638	0.0	0.000000

In [43]: # set number of clusters
kclusters = 15

```

tocluster = newdf['Hospitals/GDP']
tocluster = pd.DataFrame(tocluster)
# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(tocluster)

# check cluster labels generated for each row in the dataframe
kmeans.labels_

```

Out[43]: array([2, 8, 8, 8, 12, 2, 14, 8, 14, 5, 11, 4, 2, 8, 8, 13, 13,
 9, 6, 12, 4, 2, 11, 10, 0, 14, 9, 3, 8, 1, 7, 14, 2, 8,
 6, 7, 13, 12, 6, 12, 12, 9, 2, 10, 14, 2, 12, 2, 2, 2, 2,
 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2],
 dtype=int32)

In [44]: finalData = newdf

```
# add clustering labels
finalData['Cluster Labels'] = kmeans.labels_

finalData
```

Out[44]:

	Country	Capital	Population	\
47	Sierra Leone	Freetown	1,070,200	
42	Libya	Tripoli	1,184,045	
14	Algeria	Algiers	3,415,811	
91	Guyana	Georgetown, Guyana	134,599	
111	Tuvalu	Funafuti	4,492	
107	Belize	Belmopan	16,451	
60	Niger	Niamey	794,814	
93	Bhutan	Thimphu	101,259	
45	Armenia	Yerevan	1,080,487	
74	Paraguay	Asunción	520,722	
92	Sri Lanka	Colombo	118,556	
20	Brazil	Brasília	2,648,532	
102	Monaco	Monaco	35,986	
77	Bosnia and Herzegovina	Sarajevo	395,133	
26	Cuba	Havana	2,135,498	
83	Tanzania	Dodoma	287,200	
39	Zambia	Lusaka	1,331,254	
71	Tajikistan	Dushanbe	582,496	
88	Benin	Porto-Novo	223,552	
85	Suriname	Paramaribo	254,147	
56	Malawi	Lilongwe	902,388	
72	Eritrea	Asmara	1,258,001	
66	Honduras	Tegucigalpa	735,982	
37	Yemen	Sana'a	1,431,649	
0	China	Beijing	20,693,000	
30	Sudan	Khartoum	1,740,661	
28	Venezuela	Caracas	1,838,939	
16	Kenya	Nairobi	3,138,369	
100	Samoa	Apia	39,813	
46	Mozambique	Maputo	1,076,689	
67	Central African Republic	Bangui	731,548	
7	Bangladesh	Dhaka	8,906,000	
97	Comoros	Moroni, Comoros	60,200	
104	Brunei	Bandar Seri Begawan	28,135	
40	Mali	Bamako	1,289,626	
9	Colombia	Bogotá	7,613,303	
79	South Sudan	Juba	372,410	
11	Iraq	Baghdad	7,216,040	
69	Rwanda	Kigali	718,414	
63	Albania	Tirana	763,634	
75	Djibouti	Djibouti (city)	475,332	

54	Nicaragua	Managua	926,883
113	Palau	Ngerulmud	391
50	Liberia	Monrovia	1,010,970
59	Togo	Lomé	824,738
109	Grenada	St. George's, Grenada	7,500
96	Saint Lucia	Castries	70,000
62	Nigeria	Abuja	778,567
76	Guinea-Bissau	Bissau	452,640
99	Saint Vincent and the Grenadines	Kingstown	40,020
44	Dominican Republic	Santo Domingo	1,111,838
4	Philippines	Metro Manila	12,877,253
13	Syria	Damascus	3,500,000
35	Zimbabwe	Harare	1,487,028
17	Ethiopia	Addis Ababa	3,040,740
58	Kyrgyzstan	Bishkek	843,240
78	Burundi	Bujumbura	384,461
31	Uganda	Kampala	1,659,600
15	Afghanistan	Kabul	3,140,853
27	Cambodia	Phnom Penh	2,011,725
106	Tonga	Nukualofa	22,400
86	Namibia	Windhoek	252,721
25	Azerbaijan	Baku	2,204,200
89	Kosovo	Prishtina	198,214
3	India	New Delhi	16,787,949
48	Senegal	Dakar	1,030,594
105	Marshall Islands	Majuro	25,400
87	Botswana	Gaborone	225,656
5	Egypt	Cairo	10,230,350
36	Qatar	Doha	1,450,000
80	Costa Rica	San José, Costa Rica	328,195
101	Vanuatu	Port Vila	38,000
108	Dominica	Roseau	14,847
112	Nauru	De facto	1,100
51	Burkina Faso	Ouagadougou	1,005,231
61	Moldova	Chiinu	794,800
21	Jordan	Amman	2,600,603
110	Liechtenstein	Vaduz	5,248
33	Madagascar	Antananarivo	1,613,375
12	Singapore	Singapore	5,535,000
103	Kiribati	Tarawa	30,000
53	Pakistan	Islamabad	955,629
84	Lesotho	Maseru	267,652
23	South Africa	Pretoria	2,345,908
52	Nepal	Kathmandu	1,003,285
24	Uzbekistan	Tashkent	2,207,850
55	Mongolia	Ulaanbaatar	907,802
65	Chad	N'Djamena	751,288
64	Tunisia	Tunis	767,629

22		Angola		Luanda	2,453,779
18		Ukraine		Kiev	2,847,200
49		Guatemala		Guatemala City	1,022,000
57		Bolivia		La Paz	877,363
10		Vietnam		Hanoi	7,587,800
94		Fiji		Suva	84,410
19		Cameroon		Yaoundé	2,765,568
38		Guinea		Conakry	1,399,981
70		Jamaica		Kingston, Jamaica	701,063
95		Luxembourg		Luxembourg (city)	76,420
98		Solomon Islands		Honiara	59,288

	GDP	longitude	latitude	Number of Hospitals	Hospitals/GDP	\
47	1600	-13.231723	8.465677	0.0	0.000000	
42	10000	13.191338	32.887209	4.0	0.400000	
14	15200	3.058756	36.753768	8.0	0.526316	
91	8200	-58.155125	6.801279	4.0	0.487805	
111	3800	179.196193	-8.521147	1.0	0.263158	
107	8300	-88.759020	17.251011	1.0	0.120482	
60	1200	2.125385	13.511596	2.0	1.666667	
93	8700	89.639286	27.472792	4.0	0.459770	
45	9500	44.499103	40.179186	17.0	1.789474	
74	9800	-57.575926	-25.263740	95.0	9.693878	
92	12800	79.861243	6.927079	58.0	4.531250	
20	15600	-47.921820	-15.826691	64.0	4.102564	
102	115700	7.424616	43.738418	4.0	0.034572	
77	12700	18.413076	43.856259	6.0	0.472441	
26	12300	-82.366596	23.113592	6.0	0.487805	
83	3200	35.751607	-6.162959	4.0	1.250000	
39	4000	28.322816	-15.387526	5.0	1.250000	
71	3200	68.787038	38.559772	11.0	3.437500	
88	2300	2.628852	6.496857	2.0	0.869565	
85	14600	-55.203828	5.852036	4.0	0.273973	
56	1200	33.774119	-13.962612	5.0	4.166667	
72	1600	38.925052	15.322877	0.0	0.000000	
66	5600	-87.192136	14.072275	25.0	4.464286	
37	1300	44.191007	15.369445	2.0	1.538462	
0	16700	116.407396	39.904200	33.0	1.976048	
30	4600	32.559899	15.500654	8.0	1.739130	
28	12100	-66.903606	10.480594	42.0	3.471074	
16	3500	36.821946	-1.292066	56.0	16.000000	
100	5700	-171.751355	-13.850696	3.0	0.526316	
46	1200	32.573175	-25.969248	9.0	7.500000	
67	700	18.558190	4.394673	2.0	2.857143	
7	4200	90.412518	23.810332	7.0	1.666667	
97	1600	43.247315	-11.717216	0.0	0.000000	
104	78200	114.939821	4.903052	40.0	0.511509	
40	2200	-8.002889	12.639232	2.0	0.909091	

9	14500	-74.072092	4.710989	41.0	2.827586
79	1500	31.571250	4.859363	2.0	1.333333
11	17000	44.361488	33.312806	4.0	0.235294
69	2100	30.104429	-1.970579	2.0	0.952381
63	12500	19.818698	41.327546	4.0	0.320000
75	3600	43.145648	11.572076	1.0	0.277778
54	5800	-86.236174	12.114993	21.0	3.620690
113	16200	134.624289	7.500383	0.0	0.000000
50	1400	-10.807370	6.315607	2.0	1.428571
59	1700	1.225418	6.125626	3.0	1.764706
109	14900	-61.748800	12.056097	1.0	0.067114
96	14400	-60.987469	14.010109	5.0	0.347222
62	5900	7.398574	9.076479	0.0	0.000000
76	1800	-15.584323	11.863220	0.0	0.000000
99	11500	-61.224816	13.160025	0.0	0.000000
44	16900	-69.931212	18.486058	0.0	0.000000
4	8300	121.022256	14.609054	0.0	0.000000
13	2900	36.276528	33.513807	0.0	0.000000
35	2300	31.033510	-17.825166	0.0	0.000000
17	2200	38.757761	8.980603	0.0	0.000000
58	3700	74.569762	42.874621	0.0	0.000000
78	700	29.359878	-3.361378	0.0	0.000000
31	2400	32.582520	0.347596	0.0	0.000000
15	2000	69.207486	34.555349	0.0	0.000000
27	4000	104.928210	11.556374	0.0	0.000000
106	5600	-175.204947	-21.139342	0.0	0.000000
86	11300	17.065755	-22.560881	0.0	0.000000
25	17500	49.867092	40.409262	0.0	0.000000
89	10500	21.165503	42.662914	0.0	0.000000
3	7200	77.209021	28.613939	0.0	0.000000
48	2700	-17.467686	14.716677	0.0	0.000000
105	3400	171.185774	7.116421	0.0	0.000000
87	17800	25.923147	-24.628208	0.0	0.000000
5	12700	31.235712	30.044420	0.0	0.000000
36	124500	51.531040	25.285447	0.0	0.000000
80	16900	-84.090725	9.928069	0.0	0.000000
101	2700	168.327325	-17.733251	0.0	0.000000
108	11100	-61.379355	15.309168	0.0	0.000000
112	12200	166.931503	-0.522778	0.0	0.000000
51	1900	-1.519660	12.371428	0.0	0.000000
61	5700	28.863810	47.010453	0.0	0.000000
21	12500	35.928372	31.945367	0.0	0.000000
110	139100	9.520928	47.141030	0.0	0.000000
33	1600	47.507905	-18.879190	0.0	0.000000
12	93900	103.819836	1.352083	0.0	0.000000
103	2000	172.971662	1.451817	0.0	0.000000
53	5400	73.047885	33.684420	0.0	0.000000
84	3600	27.486923	-29.315077	0.0	0.000000

23	13500	28.229271	-25.747868	0.0	0.000000
52	2700	85.323960	27.717245	0.0	0.000000
24	6900	69.240073	41.299496	0.0	0.000000
55	13000	106.905744	47.886399	0.0	0.000000
65	2300	15.055741	12.134846	0.0	0.000000
64	11800	10.181532	36.806495	0.0	0.000000
22	6800	13.230176	-8.814656	0.0	0.000000
18	8700	30.523400	50.450100	0.0	0.000000
49	8100	-90.506882	14.634915	0.0	0.000000
57	7500	-68.119294	-16.489689	0.0	0.000000
10	6900	105.834160	21.027764	0.0	0.000000
94	9800	178.450079	-18.124809	0.0	0.000000
19	3700	11.502075	3.848032	0.0	0.000000
38	2000	-13.578401	9.641185	0.0	0.000000
70	9200	-76.809904	18.017874	0.0	0.000000
95	106300	6.131935	49.611621	0.0	0.000000
98	2200	159.972900	-9.445638	0.0	0.000000

Cluster Labels

47	2
42	8
14	8
91	8
111	12
107	2
60	14
93	8
45	14
74	5
92	11
20	4
102	2
77	8
26	8
83	13
39	13
71	9
88	6
85	12
56	4
72	2
66	11
37	10
0	0
30	14
28	9
16	3
100	8

46	1
67	7
7	14
97	2
104	8
40	6
9	7
79	13
11	12
69	6
63	12
75	12
54	9
113	2
50	10
59	14
109	2
96	12
62	2
76	2
99	2
44	2
4	2
13	2
35	2
17	2
58	2
78	2
31	2
15	2
27	2
106	2
86	2
25	2
89	2
3	2
48	2
105	2
87	2
5	2
36	2
80	2
101	2
108	2
112	2
51	2
61	2
21	2

```
110      2
33       2
12       2
103      2
53       2
84       2
23       2
52       2
24       2
55       2
65       2
64       2
22       2
18       2
49       2
57       2
10       2
94       2
19       2
38       2
70       2
95       2
98       2
```

```
In [45]: # create map
map_clusters = folium.Map(location=[0,0], zoom_start=2)

# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i+x+(i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(finalData['latitude'], finalData['longitude'], finalData['poi'], finalData['cluster']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=10,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```

Out[45]: <folium.folium.Map at 0x7f9494fd5940>

